

10 Tips for Managing CF projects

Shlomy Gantz
President, BlueBrick



INNOVATIVE SOLUTIONS, PROVEN RESULTS

www.figleaf.com

50

~~10~~ Tips for Managing CF projects

Shlomy Gantz

President, BlueBrick



FIG LEAF
SOFTWARE

INNOVATIVE SOLUTIONS, PROVEN RESULTS

www.figleaf.com

About me

```
<CFSET CurrentTitle = "President, BlueBrick Inc.">

<CFSET experience_YY = 16>
<CFSET experience_CF = 11>
<CFSET experience_PM = 12>

<CFSET aTitles = arrayNew(1)>
<CFSET aTitles[1] = "Adobe Certified Instructor">
<CFSET aTitles[2] = "Adobe Community Expert">
<CFSET aTitles[3] = "Manager, NYFLEX user group">
<CFSET aTitles[4] = "Speaker, CFUNITED, Max..">
<CFSET aTitles[5] = "Author, CF Developer's Handbook, CFDJ">

<CFSET Mom = "Very Proud">
```

Agenda

- ⌘ **Project Buzzwords & Myths**
- ⌘ **Reasons for Failure**
- ⌘ **Project Success - A Common Sense Approach**
- ⌘ **Q&A**

Project Buzzwords & Myths



INNOVATIVE SOLUTIONS, PROVEN RESULTS

www.figleaf.com

“Process”



FIG LEAF
SOFTWARE

INNOVATIVE SOLUTIONS, PROVEN RESULTS

www.figleaf.com

"Our unique process empowers our best-of-breed team, delivering results on time on budget."



FIG LEAF
SOFTWARE

SOLUTIONS. PROVEN RESULTS

www.figleaf.com

Buzzwords - "Process"

- ⌘ 4 Step, 5 Step, 12 Step...
- ⌘ Who designed your process?
- ⌘ Do you actually follow that process?
- ⌘ When did you last update your process?

“Standards”



FIG LEAF
SOFTWARE

INNOVATIVE SOLUTIONS, PROVEN RESULTS

www.figleaf.com

Standards

⌘ Do you have coding standards?

⌘ <http://livedocs.adobe.com/wtg/public/>

⌘ Do you enforce/check coding standards ?

“Requirements”



FIG LEAF
SOFTWARE

INNOVATIVE SOLUTIONS, PROVEN RESULTS

www.figleaf.com

I can't start the project until I get some user requirements

Here, I wrote them down on this napkin...

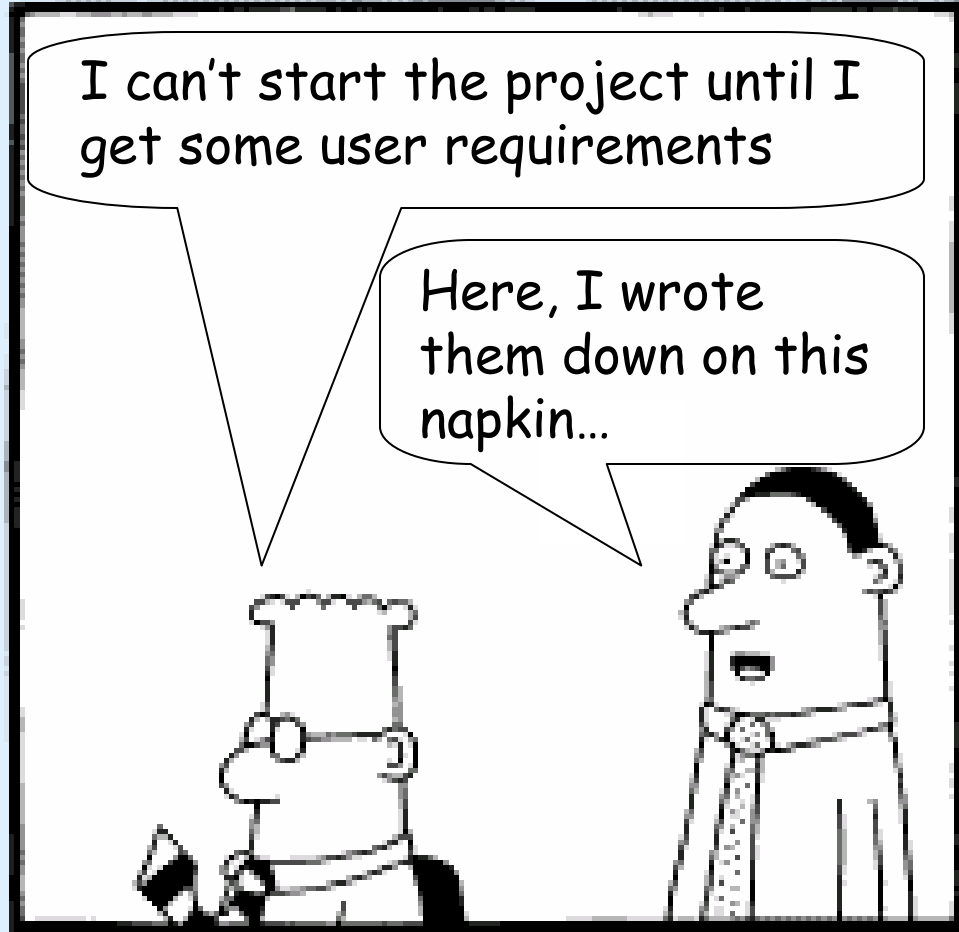


FIG LEAF
SOFTWARE

INNOVATIVE SOLUTIONS. PROVEN RESULTS

www.figleaf.com

Buzzwords – “Requirements”

- ⌘ Do you have enough requirements before you start coding?
- ⌘ Can you have too much?
- ⌘ Are they consistent?
- ⌘ Can your requirements change?
- ⌘ Are they simple to understand?
- ⌘ Are they communicated ?

"Project Managers"



INNOVATIVE SOLUTIONS, PROVEN RESULTS

www.figleaf.com

Buzzwords – Project Managers

- ⌘ **Project Managers are overrated**

- ⌘ **Project Leader vs. Project Reporter**
 - Vision
 - Determination
 - Optimism

“On Time, On Budget”



FIG LEAF
SOFTWARE

INNOVATIVE SOLUTIONS, PROVEN RESULTS

www.figleaf.com

Buzzwords – “on time and on budget...”

- ⌘ **Nearly a third of IT projects**
were **CANCELED** before
they could be completed.
- ⌘ **Over half of the projects cost**
almost **TWICE** as much as
their **original approved**
budget



INNOVATIVE SOLUTIONS, PROVEN RESULTS

www.figleaf.com

“Teamwork”



FIG LEAF
SOFTWARE

INNOVATIVE SOLUTIONS, PROVEN RESULTS

www.figleaf.com

Buzzwords – “Teamwork”

- ⌘ A development team is as strong as the strongest link.
- ⌘ Team consensus often used to avoid accountability

Reasons for Failure



FIG LEAF
SOFTWARE

INNOVATIVE SOLUTIONS, PROVEN RESULTS

www.figleaf.com

Famous *first* words

"We won't make as many mistakes on this project as we made on our last project.."

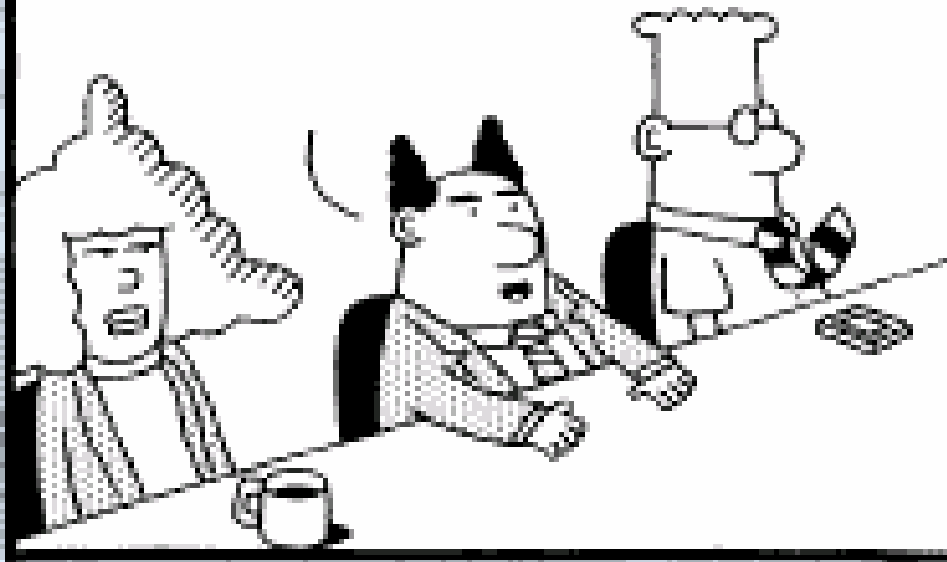


FIG LEAF
SOFTWARE

INNOVATIVE SOLUTIONS, PROVEN RESULTS

www.figleaf.com

Famous *first* words



FIG LEAF
SOFTWARE

INNOVATIVE SOLUTIONS, PROVEN RESULTS

www.figleaf.com

Famous *first* words



FIG LEAF
SOFTWARE

INNOVATIVE SOLUTIONS. PROVEN RESULTS

www.figleaf.com

Famous *first* words

"We'll be a lot more productive this time than we were last time.."

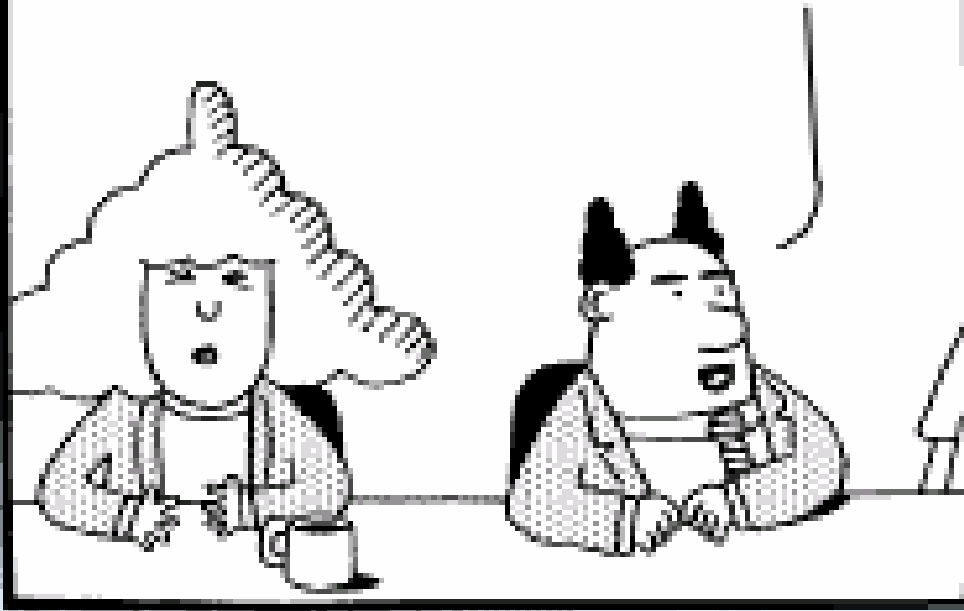


FIG LEAF
SOFTWARE

INNOVATIVE SOLUTIONS, PROVEN RESULTS

www.figleaf.com

Famous *first* words



FIG LEAF
SOFTWARE

INNOVATIVE SOLUTIONS, PROVEN RESULTS

www.figleaf.com

Reasons for Failure

- ∅ **In software**, past performance **is your *best indicator of*** future performance
- ∅ “Project success **is determined** in the **first month**”

Project Success Factors

1. User Involvement	20
2. Executive Management Support	15
3. Clear Statement of Requirements	15
4. Proper Planning	10
5. Realistic Expectations	10
6. Smaller Project Milestones	10
7. Competent Staff	5
8. Ownership	5
9. Clear Vision & Objectives	5
10. Hard-Working, Focused Staff	5

The CHAOS Report , Standish Group

Project Success



FIG LEAF
SOFTWARE

INNOVATIVE SOLUTIONS, PROVEN RESULTS

www.figleaf.com

Before You Start

- ⌘ **Define and Communicate Process**
- ⌘ **Define and Communicate Standards**
- ⌘ **Define and Communicate Resources**

Before You Start – Define Process

- ⌘ **Clearly written and communicated**
- ⌘ **Simple to follow**
 - High Level Process instead of procedures and forms
- ⌘ **Adapt your process , not your developers**

Before You Start – Define Standards

- ⌘ **Clearly written and communicated**
- ⌘ **Define Compliance and Constraints**
- ⌘ **Adjusted per Project**

- ⌘ **If it is not written, communicated and understood , it is not a standard.**



Before You Start – Define Resources

- ⌘ **Hardware List**
- ⌘ **Software List**
 - Development tools
 - Third Party Libraries
 - Version Control
 - Testing tools
 - Bug/Defect Tracking
- ⌘ **Third Party Resources**

Discover - A Clear Vision

☞ Storyboarding / Wireframes / Prototypes

- The Topic
- The Classifications
- The Specific Ideas

☞ User Personas/ Scenarios

☞ Evolving Simplified Documentation

- Wiki
- Email 2 Wiki

☞ Success Metrics

Discover – Common Mistakes

- ⌘ **Assuming the client understands you**
- ⌘ **Assuming you understand the client**
- ⌘ **Assuming the client won't change their mind**

Project Success – Plan



FIG LEAF
SOFTWARE

INNOVATIVE SOLUTIONS. PROVEN RESULTS

www.figleaf.com

Plan

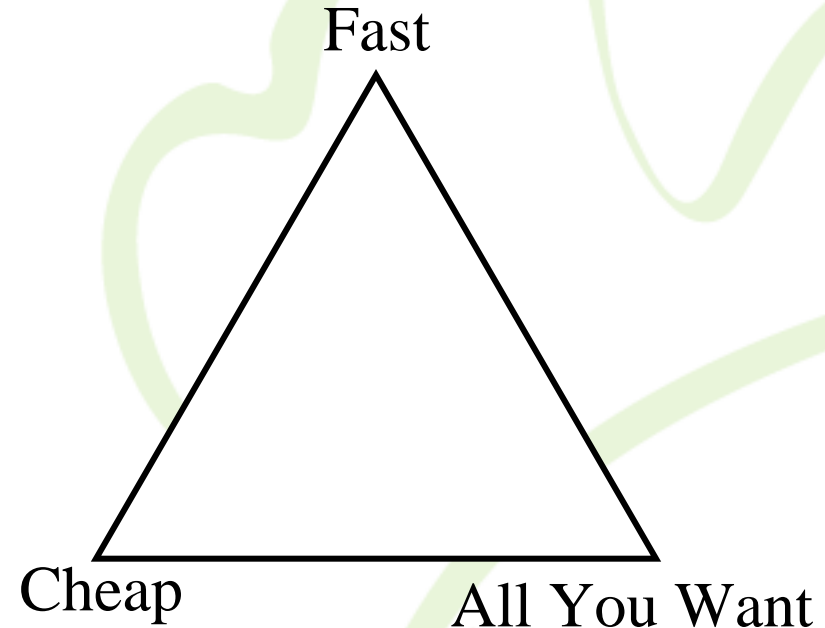
- ⌘ **Written down and communicated**
- ⌘ **Establish Communication Protocols**
- ⌘ **Includes Risks**
- ⌘ **Don't forget your test plan**
- ⌘ **Backup and Recovery Plan**



Plan – The triangle

☞ The “Iron Triangle”

- Schedule
- Cost
- Scope



☞ Your Client always wants all three

Plan – Common Mistakes

⌘ Too Much Detail

- “Create a page showing employees, using <CFQUERY>”

⌘ Not enough Detail

- “Task 1: Create Website”
- “Task 2: Collect Payment”

⌘ Project and Product summary

- Plain English
- Who’s who

Plan – What are you going to get?

⌘ Clients don't really read/react to:

- UML
- Diagrams
- Specs

⌘ Clients actually read/react to:

- Screenshots
- Prototypes

Plan – By When

⌘ Based on

- Past Experience
- Developer Feedback
- Dictated release date.

⌘ Prepare for change, “Murphy Rules”

⌘ Beware of the “Fudge Factor”

⌘ Parkinson’s Law

Plan – How you going to get there?

- ∅ Differentiate the major tasks from the little stuff
- ∅ Group minor related tasks under the major tasks.
- ∅ Sequence the major tasks into logical progression.
- ∅ Figure out who's going to do which task(s) and their constraints.
- ∅ Present project plan to team
 - Always easier to react to a plan than to build one.



Plan – Don't forget ..

⌘ The ANTI-Requirements

- Security
- Functionality

⌘ Performance Metrics

Develop – Executing the Plan

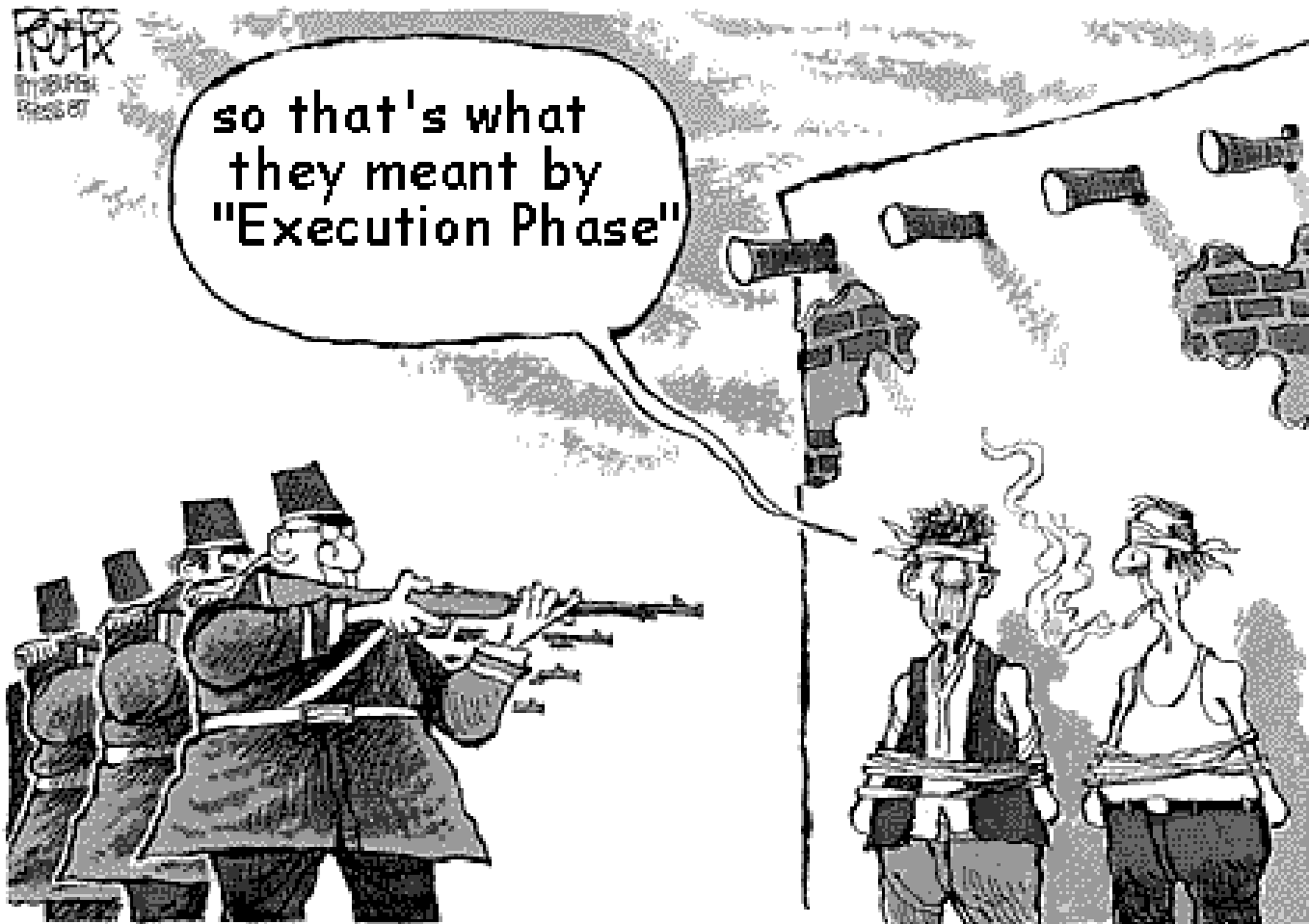


FIG LEAF
SOFTWARE

INNOVATIVE SOLUTIONS. PROVEN RESULTS

www.figleaf.com

Develop – Famous Last Words

- ⌘ “I’ll patch it for now, and create the reusable component later...”
- ⌘ “We only need to use this once...”
- ⌘ “Can you throw together a quick prototype? Don’t worry ... we won’t use it in production this time”

Develop – Cowboy Coding

“Process ?!
we don’t need no stinkin’ process”



FIG LEAF
SOFTWARE

INNOVATIVE SOLUTIONS. PROVEN RESULTS

www.figleaf.com

Develop – Common Mistakes

- ⌘ **Standards and Documentation**
- ⌘ **Code Focus vs. Product Focus**
- ⌘ **Hand Coding Syndrome**
- ⌘ **Pointless/Endless meetings**
- ⌘ **Post Production Testing**
- ⌘ **Build vs. Buy**

Develop - Standards and Documentation

- ⌘ "It's just one query...we don't need the component for that"
- ⌘ "I use shorter variable names, they make my code look better"
- ⌘ "I'll comment this code later..."
- ⌘ "Sure, we got standards... lots of them"



Develop – Documentation “Recommendations”

- ⌘ **Write all documentation in Latin**
 - Quidquid latine dictum sit, altum sonatur
- ⌘ **Only Document the obvious**

```
<!-- Looping over query here -->
```
- ⌘ **Document less to save code**
- ⌘ **Name variables after your favorite Monty Python characters**

```
<cfset killerRabbit = structNew(>
```



Code Focus vs. Product Focus

- ⌘ The “milisecond” Trap
- ⌘ Focus on the Final Result
- ⌘ Scale Hardware – Not Software
- ⌘ Consider Development Time when estimating cost

Develop - Hand Coding Syndrome

- ⌘ **Spend time doing real problem solving,**
 - Do you really want to create a new Create/Read/Update/Delete template?
- ⌘ **You cannot code faster than your computer**
- ⌘ **Get most of your project done in minimal time**

Develop Buy vs. Build

- ⌘ Do we really need another ColdFusion discussion forum?
- ⌘ Do we really need another WYSIWYG editor?
- ⌘ Do you really need to build it yourself?

Develop – Post Production Testing

- ⌘ **Don't Wait Until The End Of The Development Cycle To Test**
- ⌘ **Build And Test Often, Even Daily**
- ⌘ **Use Build Tools To Automate**
 - ant

Deployment and Maintenance

- ⌘ **Development Doesn't End At Deployment**
- ⌘ **Plan For Maintenance In Advance**
- ⌘ **Create Support Structure And Documentation**

In Summary...



FIG LEAF
SOFTWARE

INNOVATIVE SOLUTIONS, PROVEN RESULTS

www.figleaf.com

Great Success !!!



Resources



INNOVATIVE SOLUTIONS, PROVEN RESULTS

www.figleaf.com

Sites

✂ <http://www.construx.com>

Steve McConnel

✂ <http://www.shoottheprojectmanager.com>

Robert Brents

✂ <http://www.extremeprogramming.org>

✂ <http://www.agilealliance.org>



FIG LEAF
SOFTWARE

INNOVATIVE SOLUTIONS, PROVEN RESULTS

www.figleaf.com

Books

- ☞ **“The Accidental Project Manager”**
 - Patricia Ensworth
- ☞ **“Rapid Development”**
 - Steve McConnell
- ☞ **“Necessary, But Not Sufficient”**
 - E. Goldart
- ☞ **“The Software Development Edge”**
 - Joe Marasco
- ☞ **“The Mythical Man Month”**
 - Frederick P. Brooks, Jr.

Q & A

shlomy@bluebrick.com

<http://www.shlomygantz.com>



FIG LEAF
SOFTWARE

INNOVATIVE SOLUTIONS, PROVEN RESULTS

www.figleaf.com